THE CLAIMS

1. **(Original)** A method of operating upon digital data comprising the steps of:

partitioning the digital data into a plurality of blocks;

creating a plurality of first threads, such that each first thread includes at least one of the plurality of blocks; and

operating upon each of the plurality of first threads to obtain a plurality of compressed first threads, each compressed first thread including at least one compressed block of digital data.

2. **(Original)** A method according to claim 1 wherein the step of operating upon each of the first threads performs lossless compression.

3. **(Original)** A method according to claim 1 wherein the step of operating upon each of the first threads independently operates upon each of the plurality of first threads.

4. **(Original)** A method according to claim 1 wherein at least certain ones of the first threads are independently operated upon in parallel.

5. **(Original)** A method according to claim 4 wherein, during the step of operating, at least two different compression algorithms are used to independently operate upon different first threads.

6. **(Previously Amended)** A method according to claim 1 further comprising the step of combining compressed blocks in each of the plurality of compressed first threads to obtain digitally compressed data.

7. **(Original)** A method according to claim 1 wherein the step of creating the plurality of first threads includes the step of associating each of the plurality of blocks of digital data with one of the plurality of first threads such that blocks within each of the plurality of first threads share certain common compression characteristics.

8. **(Original)** A method according to claim 7 further including the step of predicting an estimated compression time and estimated compression amount for each block.

9. **(Original)** A method according to claim 8 wherein the step of creating the plurality of first threads also uses estimated compression time and estimated compression amount to determine which blocks should be associated with the same first thread.

10. **(Original)** A method according to claim 8 wherein the estimated compression time and estimated compression amount are made based upon a selected compression algorithm, and wherein the step of predicting includes the step of determining whether a proposed estimated completion time that is based upon one of the compression algorithms available for selection will allow for a desired compression amount to be achieved within a desired compression time for the digital data.

11. **(Original)** A method according to claim 1 wherein the step of creating each of the plurality of first threads uses a data type of each of the plurality of blocks so that each of the first threads contains blocks which have a similar data type.

12. **(Original)** A method according to claim 11 wherein the data type is determined according to header information related to each block.

13. **(Original)** A method according to claim 11 where the data type is determined by comparing the block data to various predetermined data patterns.

14. **(Original)** A method according to claim 1 further including the step of predicting an estimated compression time and estimated compression amount for each block.

15. **(Original)** A method according to claim 14 wherein the step of creating the plurality of first threads uses estimated compression time and estimated compression amount to determine which blocks should be associated with the same first thread.

16. **(Original)** A method according to claim 1 wherein the step of partitioning data includes the step of determining a size of each of the plurality of blocks taking data type of each block into account.

17. **(Original)** A method according to claim 1 further including the steps of:

operating upon each of the compressed first threads to eliminate each of the compressed first threads and retain the compressed first blocks;

creating a plurality of second threads, such that each second thread includes at least one of the plurality of compressed first blocks; and

operating upon each of the plurality of second threads to obtain a plurality of compressed second threads, each compressed second thread including at least one compressed second block of digital data.

18. **(Original)** A method according to claim 17 wherein the step of operating upon each of the second threads independently operates upon each of the plurality of second threads.

19. **(Original)** A method according to claim 17 wherein at least certain ones of the second threads are independently operated upon in parallel.

20. **(Original)** A method according to claim 17 wherein, during the step of operating upon each of the plurality of second threads, the same compression algorithm used to operate upon each block is also used to operate upon the corresponding compressed block.

21. **(Previously Amended)** A method according to claim 17 further comprising the step of combining the compressed blocks in each of the plurality of compressed second threads to obtain digitally compressed data.

22. **(Original)** A method according to claim 17 wherein the step of creating the plurality of second threads includes the step of associating each of the plurality of compressed first blocks with one of the plurality of second threads such that compressed first blocks within each of the plurality of second threads share certain common compression characteristics.

23. **(Original)** A method according to claim 22 wherein each of the second threads contains compressed first blocks that were created from the same first thread.

24. **(Original)** A method according to claim 23 wherein the number of second threads is greater than the number of first threads.

25. **(Original)** A method according to claim 22 wherein compressed first blocks that were within the one of the first threads are used to form two distinct second threads.

26. **(Original)** A method according to claim 17 wherein the step of operating upon each of the plurality of first threads also results in obtaining a plurality of first metadata sets, each first metadata set including portions of compressed first blocks which are determined to possibly have redundancies disposed therein.

27. **(Original)** A method according to claim 26 wherein the step of operating upon each of the first threads will maintain for each thread a pattern of data in an initial compressed first block that corresponds to a first metadata pattern, each of the different first metadata patterns for each first thread combining to result in the first metadata set for that first thread.

28. **(Original)** A method according to claim 27 wherein the pattern of data in the initial compressed first block is maintained during subsequent steps.

29. **(Original)** A method according to claim 17 wherein each first thread has an associated first metadata set.

30. **(Original)** A method according to claim 26 wherein the step of creating the plurality of second threads includes the steps of:

determining which compressed first blocks should be associated with the same second thread; and

using the first metadata sets to eliminate redundancies in some of the compressed first blocks associated with at least some of the second threads.

31. **(Original)** A method according to claim 30 wherein the step of operating upon each of the plurality of second threads also results in obtaining a plurality of second metadata sets, each second metadata set including portions of compressed second blocks which are determined to possibly have redundancies disposed therein.

32. **(Original)** A method according to claim 31 wherein the second metadata set is a subset of the first metadata set.

33. **(Original)** A method according to claim 26 wherein:

the step of operating upon each of the first threads will maintain for each thread a pattern of data in an initial compressed first block that corresponds to a first metadata pattern, each of the different first metadata patterns for each first thread combining to result in the first metadata set for that first thread;

the step of using the first metadata sets to eliminate redundancies in some of the compressed first blocks associated with at least some of the second threads maintains the pattern of data in the initial compressed first block and eliminates the pattern of data in a subsequently compressed first block.

34. **(Original)** A method according to claim 33 wherein, during the step of using the first metadata sets to eliminate redundancies in some of the compressed first blocks associated with at least some of the second threads, the pattern of data in the subsequently compressed first block is replaced with a pointer and an operation designator, thereby obtaining a plurality of compressed and reduced first blocks in each second thread.

35. **(Original)** A method according to claim 34 wherein, during the step of operating upon each of the plurality of second threads, the same compression algorithm used to operate upon each block is also used to operate upon the corresponding compressed and reduced first block to thereby obtain the compressed second blocks.

36. **(Original)** A method according to claim 35 wherein the operation designator identifies an operation used to eliminate redundancies in the first compressed data blocks, the

operation being one of an equal to comparison operation, a greater than or equal to comparison, and a less than or equal to comparison.

37. **(Original)** A method according to claim 36 wherein, during the step of operating upon each of the plurality of second threads, the comparison operation selected is adaptively determined.

38. **(Original)** A method according to claim 37 wherein the adaptive determination is made based upon the pattern of the compressed blocks as compared to representative file type patterns.

39. **(Original)** A method of operating upon digital data comprising the steps of:

compressing the digital data using multiple passes of a predetermined compression algorithm to obtain compressed digital data; and

decompressing the compressed digital data using a single pass of a corresponding decompression algorithm to obtain the digital data.

40. **(Original)** An apparatus for operating upon digital data comprising the steps of:

means for compressing the digital data using multiple passes of a predetermined compression algorithm to obtain compressed digital data; and

means for decompressing the compressed digital data using a single pass of a corresponding decompression algorithm to obtain the digital data.

41. **(Original)** An apparatus according to claim 40 wherein the means for compressing includes:

an interface controller; and

a compression engine.

42. **(Previously Amended)** An apparatus according to claim 41 wherein the compression engine comprises a single central processing unit.

43. **(Previously Amended)** An apparatus according to claim 41 wherein the compression engine comprises a plurality of central processing units.

44. **(Previously Amended)** An apparatus according to claim 43 wherein each of the plurality of central processing units operate upon a different plurality of threads.

45. **(Original)** An apparatus according to claim 44 wherein the plurality of central processing units comprise a plurality of digital signal processors.

46. **(Original)** A method of allowing a plurality of compression systems to operate more efficiently comprising the steps of:

obtaining metadata representative of patterns in first digital data obtained from the compression of the first digital data in a first compression system; and

distributing the metadata to the at least a second compression system so that the second compression system can use the metadata to compress second digital data which the second compression system needs to compress.

47. **(Previously Presented)** A method according to claim 1 wherein each first thread further includes control signals.

48. **(Previously Presented)** A method according to claim 47 wherein the control signals in each first thread include a compression routine control signal indicating a compression routine to be used in the step of operating.

49. **(Previously Presented)** A method according to claim 48 wherein different ones of the compression routine control signals indicate different compressions routines for different first threads.

50. **(Previously Presented)** A method according to claim 48 wherein different ones of the compression routine control signals indicate a same compressions routine for different first threads.

51. **(Previously Presented)** A method according to claim 47, wherein different ones of the first threads include blocks of data containing different types of data.

52. **(Previously Presented)** A method according to claim 1, wherein different ones of the first threads include blocks of data containing different types of data.

53. **(Previously Presented)** A method according to claim 7, wherein different ones of the first threads include blocks of data that do not share common compression characteristics.

54. **(Previously Presented)** A method according to claim 19 wherein at least certain ones of the first threads are independently operated upon in parallel.

55. **(Previously Presented)** A method according to claim 4 wherein each first thread further includes control signals.

56. **(Previously Presented)** A method according to claim 55 wherein the control signals in each first thread include a compression routine control signal indicating a compression routine to be used in the step of operating.

57. **(Previously Presented)** A method according to claim 56 wherein different ones of the compression routine control signals indicate different compressions routines for different first threads.

58. **(Previously Presented)** A method according to claim 56 wherein different ones of the compression routine control signals indicate a same compressions routine for different first threads.

59. **(Previously Presented)** A method according to claim 55, wherein different ones of the first threads include blocks of data containing different types of data.

60. **(Previously Presented)** A method according to claim 59 wherein, during the step of operating upon each of the plurality of first threads, at least two different compression algorithms are used to independently operate upon different first threads.

61. **(Previously Presented)** A method according to claim 39, wherein the step of compressing the digital data using multiple passes includes the step of

partitioning the digital data into a plurality of blocks;

creating a plurality of first threads, such that each first thread includes at least one of the plurality of blocks; and

operating upon each of the plurality of first threads to obtain a plurality of compressed first threads, each compressed first thread including at least one compressed block of digital data.

62. **(Previously Presented)** A method according to claim 61 wherein at least certain ones of the first threads are independently operated upon in parallel.

63. **(Previously Presented)** A method according to claim 62 wherein, during the step of operating, at least two different compression algorithms are used to independently operate upon different first threads.

64. **(Previously Presented)** A method according to claim 61, wherein the step of compressing further includes the steps of:

operating upon each of the compressed first threads to eliminate each of the compressed first threads and retain the compressed first blocks;

creating a plurality of second threads, such that each second thread includes at least one of the plurality of compressed first blocks; and

operating upon each of the plurality of second threads to obtain a plurality of compressed second threads, each compressed second thread including at least one compressed second block of digital data.

65. **(Previously Presented)** A method according to claim 64 wherein at least certain ones of the second threads are independently operated upon in parallel.

66. **(Previously Presented)** A method according to claim 64 wherein, during the step of operating upon each of the plurality of second threads, the same compression algorithm used to operate upon each block is also used to operate upon the corresponding compressed block.

67. **(Previously Presented)** A method according to claim 61 wherein each first thread has an associated first metadata set.

68. **(Previously Presented)** A method according to claim on 67 wherein each first metadata set includes a passes required variable.

69. **(Previously Presented)** A method according to claim 61 wherein each first thread further includes control signals.

70. **(Previously Presented)** A method according to claim 69 wherein the control signals in each first thread include a compression routine control signal indicating a compression routine to be used in the step of operating.

71. **(Previously Presented)** A method according to claim 70 wherein different ones of the compression routine control signals indicate different compressions routines for different first threads.

72. **(Previously Presented)** A method according to claim 70 wherein different ones of the compression routine control signals indicate a same compressions routine for different first threads.

73. **(Previously Presented)** A method according to claim 61, wherein different ones of the first threads include blocks of data containing different types of data.